

CCCCCCCCCCCC	00000000	NNN	NNN	VVV	VVV
CCCCCCCCCCCC	00000000	NNN	NNN	VVV	VVV
CCCCCCCCCCCC	00000000	NNN	NNN	VVV	VVV
CCC	000	000	NNN	VVV	VVV
CCC	000	000	NNN	VVV	VVV
CCC	000	000	NNN	VVV	VVV
CCC	000	000	NNN	VVV	VVV
CCC	000	000	NNNNNN	VVV	VVV
CCC	000	000	NNNNNN	VVV	VVV
CCC	000	000	NNNNNN	VVV	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCC	000	000	NNN	NNN	VVV
CCCCCCCCCCCC	00000000	NNN	NNN	VVV	VVV
CCCCCCCCCCCC	00000000	NNN	NNN	VVV	VVV
CCCCCCCCCCCC	00000000	NNN	NNN	VVV	VVV

RRRRRRR RRRPRRR RR RR RR RR RR RR RRRRRRR RRRRRRR RR RR RR RR RR RR RR RR RR RR RR RR	EEEEEEEE EEEEEEEE EE EE EE EEEEEEEE EEEEEEEE EE EE EE EEEEEEEE EEEEEEEE	CCCCCCCC CCCCCCCC CC CC CC CC CC CC CC CC CC CCCCCCCC CCCCCCCC	LL LL LL LL LL LL LL LL LL LL LL LLLLLLLLLL LLLLLLLLLL	DDDDDDDD DDDDDDDD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DDDDDDDD DDDDDDDD	CCCCCCCC CCCCCCCC CC CC CC CC CC CC CC CC CCCCCCCC CCCCCCCC	LL LL LL LL LL LL LL LL LL LL LL LLLLLLLLLL LLLLLLLLLL
---	--	--	--	---	--	--	------------------------------

LL LL LL LL LL LL LL LL LL LL LLLLLLLLLL LLLLLLLLLL	IIIIII IIIIII II II II II II II II II II II II II II II II II II	SSSSSSSS SSSSSSSS SS SS SS SS SSSSSS SSSSSS SS SS SS SS SSSSSS SSSSSS
--	--	--

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

```

0001 0 %TITLE 'VAX-11 CONVERT/RECLAIM'
0002 0 MODULE RECLSDCL ( IDENT='V04-000',
0003 0 MAIN=START
0004 0 ) =
0005 0
0006 1 BEGIN
0007 1
0008 1 !*****
0009 1 !
0010 1 ! * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0011 1 ! * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0012 1 ! * ALL RIGHTS RESERVED. *
0013 1 !
0014 1 ! * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0015 1 ! * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0016 1 ! * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0017 1 ! * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0018 1 ! * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0019 1 ! * TRANSFERRED. *
0020 1 !
0021 1 ! * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0022 1 ! * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0023 1 ! * CORPORATION. *
0024 1 !
0025 1 ! * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0026 1 ! * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0027 1 !
0028 1 !
0029 1 !*****

```

```

31 0030 1 |++
32 0031 1 |
33 0032 1 | Facility: VAX-11 CONVERT/RECLAIM
34 0033 1 |
35 0034 1 | Environment:
36 0035 1 |
37 0036 1 | VAX/VMS Operating System
38 0037 1 |
39 0038 1 | Abstract: Main routines
40 0039 1 |
41 0040 1 | Contents:
42 0041 1 | main
43 0042 1 |
44 0043 1 | --
45 0044 1 |
46 0045 1 |
47 0046 1 | Author: Keith B Thompson
48 0047 1 | Peter Lieberwirth Creation date: August-1981
49 0048 1 |
50 0049 1 |
51 0050 1 | Modified by:
52 0051 1 |
53 0052 1 | V03-003 KBT0373 Keith B. Thompson 20-Oct-1982
54 0053 1 | Remove the flags ref. stuff and use the new flags
55 0054 1 | parameters on the call interface
56 0055 1 |
57 0056 1 | V03-002 KBT0043 Keith Thompson 3-Apr-1982
58 0057 1 | Change the refrence to conv$ab_flags to use the vector stuff
59 0058 1 |
60 0059 1 | V03-001 KBT0020 Keith Thompson 23-Mar-1982
61 0060 1 | Correct the display of CPU time
62 0061 1 |
63 0062 1 |****

```

```

: 65 0063 1
: 66 0064 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
: 67 0065 1 REQUIRE 'SRCS:CONVERT';
: 68 0300 1 REQUIRE 'SRCS:CONVDEF';
: 69 0461 1
: 70 0462 1 EXTERNAL ROUTINE
: 71 0463 1 LIB$INIT_TIMER : ADDRESSING_MODE ( GENERAL ),
: 72 0464 1 CLISGET_VALUE : ADDRESSING_MODE ( GENERAL ),
: 73 0465 1 CLISPRESENT : ADDRESSING_MODE ( GENERAL ),
: 74 0466 1 CONVSRECLAIM : ADDRESSING_MODE ( GENERAL ),
: 75 0467 1 LIB$STAT_TIMER : ADDRESSING_MODE ( GENERAL ),
: 76 0468 1 LIB$PUT_OUTPUT : ADDRESSING_MODE ( GENERAL );
: 77 0469 1
: 78 0470 1 FORWARD ROUTINE
: 79 0471 1 MULQ : NOVALUE;
: 80 0472 1
: 81 0473 1 OWN
: 82 0474 1 FILE_NAME : DESC_BLK
: 83 0475 1 PRESET( [ DSCSB_CLASS ] = DSCSK_CLASS_D ),
: 84 0476 1
: 85 0477 1 STATISTICS_BLOCK : VECTOR [ 5, LONG ] INITIAL ( 4, 0, 0, 0, 0 ),
: 86 0478 1
: 87 0479 1 FLAGS : LONG INITIAL( CONVSM_SIGNAL ),
: 88 0480 1
: 89 0481 1 ! FAO Processing
: 90 0482 1 !
: 91 0483 1 FAO_BUFFER : VECTOR [ 132, BYTE ],
: 92 0484 1 FAO_DESC : DESC_BLK
: 93 0485 1 PRESET( [ DSCSB_CLASS ] = DSCSK_CLASS_D,
: 94 0486 1 [ DSCSW_LENGTH ] = 132,
: 95 0487 1 [ DSCSA_POINTER ] = FAO_BUFFER ),
: 96 0488 1 PUT_DESC : DESC_BLK
: 97 0489 1 PRESET( [ DSCSB_CLASS ] = DSCSK_CLASS_D,
: 98 0490 1 [ DSCSW_LENGTH ] = 132,
: 99 0491 1 [ DSCSA_POINTER ] = FAO_BUFFER ),
100 0492 1
101 0493 1 TIMER_BLK : LONG,
102 0494 1
103 0495 1 ELP_TIME : VECTOR [ 2, LONG ],
104 0496 1 CPU_TIME : VECTOR [ 2, LONG ],
105 0497 1
106 0498 1 ELP_TIM_BUF : VECTOR [ 16, BYTE ],
107 0499 1 CPU_TIM_BUF : VECTOR [ 16, BYTE ],
108 0500 1
109 0501 1 ELP_DESC : DESC_BLK INITIAL ( 16, ELP_TIM_BUF ),
110 0502 1 CPU_DESC : DESC_BLK INITIAL ( 16, CPU_TIM_BUF ),
111 0503 1
112 0504 1 ONE : INITIAL(1),
113 0505 1 TWO : INITIAL(2);
114 0506 1
115 0507 1 BIND
116 0508 1 STATS = UPLIT(
117 0509 1
118 0510 1 DESCRIPTOR( ' !/ RECLAIM Statistics' ),
119 0511 1 DESCRIPTOR( 'Total Buckets Scanned: !8UL' ),
120 0512 1 DESCRIPTOR( 'Data Buckets Reclaimed: !8UL' ),
: 121 0513 1 DESCRIPTOR( 'Index Buckets Reclaimed: !8UL' ),

```

RECLSDCL
V04-000

VAX-11 CONVERT

G 11
15-Sep-1984 23:56:35
14-Sep-1984 12:14:03

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[CONV.SRC]RECLDCL.B32;1 Page 4 (3)

```
: 122      0514 1      DESCRIPTOR( 'Total Buckets Reclaimed: !BUL' ),  
: 123      0515 1      DESCRIPTOR( 'Elapsed Time: !AS!_CPU Time: !AS' )  
: 124      0516 1  
: 125      0517 1      ) : VECTOR;  
: 126      0518 1
```

```

: 128 0519 1 %SBTTL 'Main Routine'
: 129 0520 1 ROUTINE START =
: 130 0521 1 ++
: 131 0522 1
: 132 0523 1 Functional Description:
: 133 0524 1
: 134 0525 1 DCL executable inage
: 135 0526 1
: 136 0527 1 Calling Sequence:
: 137 0528 1
: 138 0529 1 By dcl
: 139 0530 1
: 140 0531 1 Input Parameters:
: 141 0532 1 none
: 142 0533 1
: 143 0534 1 Implicit Inputs:
: 144 0535 1 none
: 145 0536 1
: 146 0537 1 Output Parameters:
: 147 0538 1 none
: 148 0539 1
: 149 0540 1 Implicit Outputs:
: 150 0541 1 none
: 151 0542 1
: 152 0543 1 Routine Value:
: 153 0544 1 none
: 154 0545 1
: 155 0546 1 Routines Called:
: 156 0547 1
: 157 0548 1 LIB$INIT_TIMER
: 158 0549 1 CLISP$PRESENT
: 159 0550 1 CLISP$GET_VALUE
: 160 0551 1 CONVS$RECLAIM
: 161 0552 1 LIB$STAT_TIMER
: 162 0553 1 $ASCTIM
: 163 0554 1 $FAO
: 164 0555 1 LIB$PUT_OUTPUT
: 165 0556 1
: 166 0557 1 Side Effects:
: 167 0558 1 none
: 168 0559 1
: 169 0560 1 --
: 170 0561 1
: 171 0562 2 BEGIN
: 172 0563 2
: 173 0564 2 BUILTIN
: 174 0565 2 SUBM;
: 175 0566 2
: 176 0567 2 LOCAL
: 177 0568 2 STATISTICS,
: 178 0569 2 LENGTH,
: 179 0570 2 TOTAL_COUNT;
: 180 0571 2
: 181 0572 2 ! Start the timer
: 182 0573 2 !
: 183 0574 2 LIB$INIT_TIMER( TIMER_BLK );
: 184 0575 2

```

```

: 185      0576      |  | Get the switch from DCL
: 186      0577      |  |
: 187      0578      |  | STATISTICS = CLISPRESNT( DESCRIPTOR( 'STATISTICS' ) );
: 188      0579      |  |
: 189      0580      |  | ! Get the input file name
: 190      0581      |  |
: 191      0582      |  | CLISGET_VALUE( DESCRIPTOR( 'FILE_NAME' ),FILE_NAME );
: 192      0583      |  |
: 193      0584      |  | ! Reclaim the file
: 194      0585      |  |
: 195      0586      |  | RET_ON_ERROR( CONV$RECLAIM ( FILE_NAME,STATISTICS_BLOCK,FLAGS ) );
: 196      0587      |  |
: 197      0588      |  | ! Output some stats if wanted
: 198      0589      |  |
: 199      0590      |  | IF .STATISTICS
: 200      0591      |  | THEN
: 201      0592      |  | BEGIN
: 202      0593      |  |
: 203      0594      |  | OWN
: 204      0595      |  |     ZERO_Q      : VECTOR [ 2,LONG ] INITIAL( 0,0 ),      ! Used for
: 205      0596      |  |     TEMP_TIME   : VECTOR [ 2,LONG ],                      ! conversion
: 206      0597      |  |     MUL100K     : VECTOR [ 2,LONG ] INITIAL( 10000,0 ); ! of times
: 207      0598      |  |
: 208      0599      |  | ! Get Preformance Stats
: 209      0600      |  |
: 210      0601      |  | LIB$STAT_TIMER( ONE,     ELP_TIME,     TIMER_BLK );
: 211      0602      |  | LIB$STAT_TIMER( TWO,    TEMP_TIME,    TIMER_BLK );
: 212      0603      |  |
: 213      0604      |  | ! Convert to delta time
: 214      0605      |  |
: 215      0606      |  | SUBM( 2,ELP_TIME,ZERO_Q,ELP_TIME );
: 216      0607      |  |
: 217      0608      |  | ! Convert internal times to ASCII
: 218      0609      |  |
: 219      P 0610      |  | $ASCTIM( TIMLEN = 0,
: 220      P 0611      |  |         TIMBUF = ELP_DESC,
: 221      P 0612      |  |         TIMADR = ELP_TIME,
: 222      0613      |  |         CVTFLG = 0 );
: 223      0614      |  |
: 224      0615      |  | ! The CPU time is given in 10msec ticks so we need to convert it to
: 225      0616      |  | system delta time
: 226      0617      |  |
: 227      0618      |  | ! Convert to 10nsec ticks
: 228      0619      |  |
: 229      0620      |  | MULQ( TEMP_TIME,MUL100K,CPU_TIME );
: 230      0621      |  |
: 231      0622      |  | ! Convert to delta time
: 232      0623      |  |
: 233      0624      |  | SUBM( 2,CPU_TIME,ZERO_Q,CPU_TIME );
: 234      0625      |  |
: 235      0626      |  | ! Conver to ascii
: 236      0627      |  |
: 237      P 0628      |  | $ASCTIM( TIMLEN = 0,
: 238      P 0629      |  |         TIMBUF = CPU_DESC,
: 239      P 0630      |  |         TIMADR = CPU_TIME,
: 240      0631      |  |         CVTFLG = 0 );
: 241      0632      |  |

```



```

: 242 0633 ! Loop to output the first 5 lines of the display
: 243 0634 !
: 244 0635 INCR I FROM 0 TO 4 BY 1
: 245 0636 DO
: 246 0637 BEGIN
: 247 0638 !
: 248 0639 ! FAO the line
: 249 0640 !
: 250 0641 $FAO( .STATS [ .I ],LENGTH,FAO_DESC,.STATISTICS_BLOCK [ .I ] );
: 251 0642 !
: 252 0643 PUT_DESC [ DSCSW_LENGTH ] = .LENGTH;
: 253 0644 !
: 254 0645 ! Output the line
: 255 0646 !
: 256 0647 LIB$PUT_OUTPUT( PUT_DESC )
: 257 0648 !
: 258 0649 END;
: 259 0650 !
: 260 0651 ! Elapsed Time and CPU Time
: 261 0652 !
: 262 0653 $FAO( .STATS [ 5 ],LENGTH,FAO_DESC,ELP_DESC,CPU_DESC );
: 263 0654 PUT_DESC [ DSCSW_LENGTH ] = .LENGTH;
: 264 0655 LIB$PUT_OUTPUT( PUT_DESC );
: 265 0656 !
: 266 0657 END;
: 267 0658 RETURN SSS_NORMAL
: 268 0659
: 269 0660
: 270 0661 END;

```

															.TITLE	RECLSDCL	VAX-11	CONVERT	
															.IDENT	\V04-000\			
															.PSECT	\$SPLITS,	NOWRT,	NOEXE,2	
61	74	53	20	4D	49	41	4C	43	45	52	20	2F	21	20	00000	P.AAC:	.ASCII	\ !/ RECLAIM Statistics\	
								73	63	69	74	73	69	74	0000F				
															00016		.BLKB	2	
															00000016	P.AAB:	.LONG	22	
															00000000		.ADDRESS	P.AAC	
53	20	73	74	65	68	63	75	42	20	6C	61	74	6F	54	00020	P.AAE:	.ASCII	\Total Buckets Scanned:	!8UL\
55	38	21	20	20	20	20	20	3A	64	65	6E	6E	61	63	0002F				
															0003E				
															0003F		.BLKB	1	
															0000001F	P.AAD:	.LONG	31	
															00000000		.ADDRESS	P.AAE	
65	52	20	73	74	65	68	63	75	42	20	61	74	61	44	00048	P.AAG:	.ASCII	\Data Buckets Reclaimed:	!8UL\
55	38	21	20	20	20	20	3A	64	65	6D	69	61	6C	63	00057				
															00066				
															00067		.BLKB	1	
															0000001F	P.AAF:	.LONG	31	
															00000000		.ADDRESS	P.AAG	
52	20	73	74	65	68	63	75	42	20	78	65	64	6E	49	00070	P.AAI:	.ASCII	\Index Buckets Reclaimed:	!8UL\
55	38	21	20	20	20	3A	64	65	6D	69	61	6C	63	65	0007F				
															0008E				
															0008F		.BLKB	1	

```

0000001F 00090 P.AAH: .LONG 31
00000000' 00094 .ADDRESS P.AAI
52 20 73 74 65 68 63 75 42 20 6C 61 74 6F 54 00098 P.AAK: .ASCII \Total Buckets Reclaimed: !8UL\
55 38 21 20 20 20 3A 64 65 6D 69 61 6C 63 65 000A7
4C 000B6
000B7
0000001F 000B8 P.AAJ: .BLKB 1
00000000' 000BC .LONG 31
20 20 3A 65 6D 69 54 20 64 65 73 70 61 6C 45 000C0 P.AAM: .ADDRESS P.AAK
69 54 20 55 50 43 5F 21 53 41 21 20 20 20 20 20 20 000C0 P.AAM: .ASCII \Elapsed Time: !AS!_CPU Time: !\
21 20 20 20 20 20 20 20 20 20 20 20 3A 65 6D 000CF
53 41 000DE
000E8
000EA
0000002A 000EC P.AAL: .BLKB 2
00000000' 000F0 .LONG 42
00000000' 000F4 P.AAA: .ADDRESS P.AAM
00000000' 000F4 P.AAA: .ADDRESS P.AAB, P.AAD, P.AAF, P.AAH, P.AAJ, P.AAL
53 43 49 54 53 49 54 41 54 53 0010C P.AAO: .ASCII \STATISTICS\
00116
0000000A 00118 P.AAN: .BLKB 2
00000000' 0011C .LONG 10
45 4D 41 4E 5F 45 4C 49 46 00120 P.AAQ: .ADDRESS P.AAO
00129
00000009 0012C P.AAP: .ASCII \FILE_NAME\
00000000' 00130 .BLKB 3
.LONG 9
.ADDRESS P.AAQ
.PSECT $OWNS,NOEXE,2

```

```

00# 00000 FILE_NAME:
02 00003 .BYTE 0[3]
00004 .BYTE 2
00000000 00000000 00000000 00000000 00000004 00008 STATISTICS_BLOCK:
00000001 0001C FLAGS: .LONG 4, 0, 0, 0, 0
00020 FAO_BUFFER: .LONG 1
0084 000A4 FAO_DESC: .BLKB 132
00 000A6 .WORD 132
02 000A7 .BYTE 0
00000000' 000A8 .BYTE 2
0084 000AC PUT_DESC: .ADDRESS FAO_BUFFER
00 000AE .WORD 132
02 000AF .BYTE 0
00000000' 000B0 .BYTE 2
000B4 TIMER_BLK: .ADDRESS FAO_BUFFER
000B8 ELP_TIME: .BLKB 4
000C0 CPU_TIME: .BLKB 8
000C8 ELP_TIM_BUF: .BLKB 8
000D8 CPU_TIM_BUF: .BLKB 16
00000010 000E8 ELP_DESC: .BLKB 16

```

```

                                .LONG 16
                                .ADDRESS ELP_TIM_BUF
00000000' 000EC CPU_DESC:
00000010 000F0
                                .LONG 16
                                .ADDRESS CPU_TIM_BUF
00000000' 000F4
00000001 000F8 ONE:
00000002 000FC TWO:
00000000 00100 ZERO_Q:
00108 TEMP_TIME:
                                .BLKB 8
00000000 000186A0 00110 MUL100K: .LONG 100000, 0
STATS=
                                P.AAA
                                .EXTRN LIB$INIT_TIMER, CLISGET_VALUE
                                .EXTRN CLISPRESENT, CONV$RECLAIM
                                .EXTRN LIB$STAT_TIMER, LIB$PUT_OUTPUT
                                .EXTRN SYSSASCTIM, SYSS$FAO
                                .PSECT $CODE$,NOWRT,2
                                00FC 00000 START:
57 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7
56 00000000G 00 9E 00009 MOVAB LIB$PUT_OUTPUT, R7
55 00000000G 00 9E 00010 MOVAB SYSS$FAO, R6
54 00000000G 00 9E 00017 MOVAB SYSSASCTIM, R5
53 0000' 0000' CF 9E 0001E MOVAB LIB$STAT_TIMER, R4
5E 0000' 0000' 04 C2 00023 MOVAB ELP_TIME, R3
                                FC A3 9F 00026 SUBL2 #4,-SP
                                0000' CF 9F 00029 PUSHAB TIMER_BLK
00000000G 00 01 FB 00029 CALLS #1, LIB$INIT_TIMER
                                0000' CF 9F 00030 PUSHAB P.AAN
00000000G 00 01 FB 00034 CALLS #1, CLISPRESENT
52 0000' 0000' 50 D0 0003B MOVL R0, STATISTICS
                                FF48 C3 9F 0003E PUSHAB FILE_NAME
00000000G 00 0000' CF 9F 00042 PUSHAB P.AAP
                                FF64 C3 9F 00046 CALLS #2, CLISGET_VALUE
                                FF50 C3 9F 0004D PUSHAB FLAGS
                                FF48 C3 9F 00051 PUSHAB STATISTICS_BLOCK
00000000G 00 0000' 03 FB 00055 PUSHAB FILE_NAME
01 0000' 0000' 50 E8 00059 CALLS #3, CONV$RECLAIM
                                03 04 00063 BLBS STATUS, 1$
                                00A2 52 E8 00064 1$: BLBS STATISTICS, 2$
                                FC A3 9F 0006A 2$: BRW 4$
                                40 A3 9F 0006D PUSHAB TIMER_BLK
64 0000' 0000' 53 DD 0006D PUSHL R3
                                44 A3 9F 0006F PUSHAB ONE
                                FC A3 9F 00072 CALLS #3, LIB$STAT_TIMER
                                50 A3 9F 00075 PUSHAB TIMER_BLK
                                44 A3 9F 00078 PUSHAB TEMP_TIME
64 0000' 0000' 03 FB 0007B PUSHAB TWO
63 48 0000' 0000' 03 FB 0007E CALLS #3, LIB$STAT_TIMER
                                50 4C A3 D0 00081 SUBL3 ELP_TIME, ZERO_Q, ELP_TIME
                                50 04 A3 D9 0008A MOVL ZERO_Q+4, R0
                                7E D4 00092 SBWC ELP_TIME+4, R0
                                53 DD 00094 MOVL R0, ELP_TIME+4
                                CLRL -(SP)
                                PUSHL R3

```

			30	A3	9F	00096		PUSHAB	ELP_DESC		
				7E	D4	00099		CLRL	-(SP)		
		65		04	FB	0009B		CALLS	#4, SYSSASCTIM		
			08	A3	9F	0009E		PUSHAB	CPU_TIME		0620
			58	A3	9F	000A1		PUSHAB	MULTOOK		
			50	A3	9F	000A4		PUSHAB	TEMP_TIME		
		0000V		03	FB	000A7		CALLS	#3, MULQ		
08	A3	48		A3	C3	000AC		SUBL3	CPU_TIME, ZERO_Q, CPU_TIME		0624
				50	A3	D0	000B3	MOVL	ZERO_Q+4, R0		
				50	A3	D9	000B7	SBWC	CPU_TIME+4, R0		
		0C		A3	D0	000BB		MOVL	R0, CPU_TIME+4		
				7E	D4	000BF		CLRL	-(SP)		0631
			08	A3	9F	000C1		PUSHAB	CPU_TIME		
			38	A3	9F	000C4		PUSHAB	CPU_DESC		
				7E	D4	000C7		CLRL	-(SP)		
		65		04	FB	000C9		CALLS	#4, SYSSASCTIM		
				52	D4	000CC		CLRL	I		0635
			FF50	C342	DD	000CE	3\$:	PUSHL	STATISTICS_BLOCK[I]		0641
			EC	A3	9F	000D3		PUSHAB	FAO_DESC		
			08	AE	9F	000D6		PUSHAB	LENGTH		
		0000'		CF42	DD	000D9		PUSHL	STATS[I]		
		66		04	FB	000DE		CALLS	#4, SYSSFAO		
		F4		6E	B0	000E1		MOVW	LENGTH, PUT_DESC		0643
				A3	9F	000E5		PUSHAB	PUT_DESC		0647
		67		01	FB	000E8		CALLS	#1, LIB\$PUT_OUTPUT		
DF		52		04	F3	000EB		AOBLEQ	#4, I, 3\$		
				A3	9F	000EF		PUSHAB	CPU_DESC		0653
				A3	9F	000F2		PUSHAB	ELP_DESC		
				A3	9F	000F5		PUSHAB	FAO_DESC		
				AE	9F	000F8		PUSHAB	LENGTH		
		0000'		CF	DD	000FB		PUSHL	STATS+20		
		66		05	FB	000FF		CALLS	#5, SYSSFAO		
		F4		6E	B0	00102		MOVW	LENGTH, PUT_DESC		0654
				A3	9F	00106		PUSHAB	PUT_DESC		0655
		67		01	FB	00109		CALLS	#1, LIB\$PUT_OUTPUT		
		50		01	D0	0010C	4\$:	MOVL	#1, R0		0659
				04	0010F			RET			0661

; Routine Size: 272 bytes, Routine Base: \$CODES + 0000

```

: 272 0662 1 %SBTTL 'MULQ'
: 273 0663 1 ROUTINE MULQ ( MUL1 : REF VECTOR [ 2, LONG ],
: 274 0664 1           MUL2 : REF VECTOR [ 2, LONG ],
: 275 0665 1           PROD : REF VECTOR [ 2, LONG ] ) : NOVALUE =
: 276 0666 1
: 277 0667 1 ++
: 278 0668 1
: 279 0669 1 Functional Description:
: 280 0670 1
: 281 0671 1     Multiplies two quadwords. This routine was converted from the example
: 282 0672 1     of the EMUL instruction in the VAX Architecture Handbook
: 283 0673 1
: 284 0674 1 Calling Sequence:
: 285 0675 1
: 286 0676 1     MULQ( mul1,mul2,prod )
: 287 0677 1
: 288 0678 1 Input Parameters:
: 289 0679 1
: 290 0680 1     mul1    - quadword multiplier
: 291 0681 1     mul2    - quadword multiplier
: 292 0682 1
: 293 0683 1 Implicit Inputs:
: 294 0684 1     none
: 295 0685 1
: 296 0686 1 Output Parameters:
: 297 0687 1
: 298 0688 1     prod    - quadword product (note: output cannot be same as either input)
: 299 0689 1
: 300 0690 1 Implicit Outputs:
: 301 0691 1     none
: 302 0692 1
: 303 0693 1 Routine Value:
: 304 0694 1     none
: 305 0695 1
: 306 0696 1 Routines Called:
: 307 0697 1     none
: 308 0698 1
: 309 0699 1 Side Effects:
: 310 0700 1     none
: 311 0701 1
: 312 0702 1 --
: 313 0703 1
: 314 0704 1 BEGIN
: 315 0705 1
: 316 0706 1 BUILTIN
: 317 0707 1     EMUL;
: 318 0708 1
: 319 0709 1 BIND
: 320 0710 1     MUL1S = MUL1 [ 0 ] : SIGNED,
: 321 0711 1     MUL2S = MUL2 [ 0 ] : SIGNED;
: 322 0712 1
: 323 0713 1 LOCAL
: 324 0714 1     ZERO : INITIAL( 0 ),
: 325 0715 1     TEMP;
: 326 0716 1
: 327 0717 1 ! Multiply low half
: 328 0718 1

```

```

: 329      0719 2      EMUL( .MUL1,.MUL2,ZERO,.PROD );
: 330      0720      ! High half = A[high] * B[low] + A[low] * B[high]
: 331      0721      !
: 332      0722      !
: 333      0723      TEMP = ( .MUL1 [ 1 ] * .MUL2 [ 0 ] ) + ( .MUL1 [ 0 ] * .MUL2 [ 1 ] );
: 334      0724      !
: 335      0725      ! If A[low]<0 then compensate of unsigned bias of 2**32
: 336      0726      !
: 337      0727      IF .MUL1S LSS 0
: 338      0728      THEN
: 339      0729      TEMP = .TEMP + .MUL2 [ 0 ];
: 340      0730      !
: 341      0731      ! If B[low]<0 then compensate of unsigned bias of 2**32
: 342      0732      !
: 343      0733      IF .MUL2S LSS 0
: 344      0734      THEN
: 345      0735      TEMP = .TEMP + .MUL1 [ 0 ];
: 346      0736      !
: 347      0737      ! Combine with high half of A[low] * B[low]
: 348      0738      !
: 349      0739      PROD [ 1 ] = .PROD [ 1 ] + .TEMP;
: 350      0740
: 351      0741      RETURN
: 352      0742
: 353      0743      END;

```

				001C 00000	MULQ:	.WORD	Save R2,R3,R4	: 0663
		53	04	AC D0 00002		MOVL	MUL1, R3	: 0710
		52	08	AC D0 00006		MOVL	MUL2, R2	: 0711
				51 D4 0000A		CLRL	ZERO	
		50	0C	AC D0 0000C		MOVL	PROD, R0	: 0719
60	51	62		63 7A 00010		EMUL	(R3), (R2), ZERO, (R0)	
	54	A3	04	62 C5 00015		MULL3	(R2), 4(R3), R4	: 0723
	51	63	04	A2 C5 0001A		MULL3	4(R2), (R3), R1	
		51		54 C0 0001F		ADDL2	R4, TEMP	
				63 D5 00022		TSTL	(R3)	: 0727
				03 18 00024		BGEQ	1\$	
		51		62 C0 00026		ADDL2	(R2), TEMP	: 0729
				62 D5 00029	1\$:	TSTL	(R2)	: 0733
				03 18 0002B		BGEQ	2\$	
		51		63 C0 0002D		ADDL2	(R3), TEMP	: 0735
	04	A0		51 C0 00030	2\$:	ADDL2	TEMP, 4(R0)	: 0739
				04 00034		RET		: 0743

: Routine Size: 53 bytes, Routine Base: \$CODE\$ + 0110

```

: 354      0744 1
: 355      0745 0 END      ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	280	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	308	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	325	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]LIB.L32;1	18619	9	0	1000	00:01.8

COMMAND QUALIFIERS

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

: Size: 325 code + 588 data bytes
 : Run Time: 00:10.9
 : Elapsed Time: 00:38.6
 : Lines/CPU Min: 4097
 : Lexemes/CPU-Min: 21195
 : Memory Used: 125 pages
 : Compilation Complete

Terminal window 1	Terminal window 2	Terminal window 3	Terminal window 4	Terminal window 5	Terminal window 6	Terminal window 7	Terminal window 8	Terminal window 9	Terminal window 10
Terminal window 11	Terminal window 12	Terminal window 13	Terminal window 14	Terminal window 15	Terminal window 16	Terminal window 17	Terminal window 18	Terminal window 19	Terminal window 20
Terminal window 21	Terminal window 22	Terminal window 23	Terminal window 24	Terminal window 25	Terminal window 26	Terminal window 27	Terminal window 28	Terminal window 29	Terminal window 30
Terminal window 31	Terminal window 32	Terminal window 33	Terminal window 34	Terminal window 35	Terminal window 36	Terminal window 37	Terminal window 38	Terminal window 39	Terminal window 40
Terminal window 41	Terminal window 42	Terminal window 43	Terminal window 44	Terminal window 45	Terminal window 46	Terminal window 47	Terminal window 48	Terminal window 49	Terminal window 50
Terminal window 51	Terminal window 52	Terminal window 53	Terminal window 54	Terminal window 55	Terminal window 56	Terminal window 57	Terminal window 58	Terminal window 59	Terminal window 60
Terminal window 61	Terminal window 62	Terminal window 63	Terminal window 64	Terminal window 65	Terminal window 66	Terminal window 67	Terminal window 68	Terminal window 69	Terminal window 70
Terminal window 71	Terminal window 72	Terminal window 73	Terminal window 74	Terminal window 75	Terminal window 76	Terminal window 77	Terminal window 78	Terminal window 79	Terminal window 80
Terminal window 81	Terminal window 82	Terminal window 83	Terminal window 84	Terminal window 85	Terminal window 86	Terminal window 87	Terminal window 88	Terminal window 89	Terminal window 90
Terminal window 91	Terminal window 92	Terminal window 93	Terminal window 94	Terminal window 95	Terminal window 96	Terminal window 97	Terminal window 98	Terminal window 99	Terminal window 100