



\*\*FILE\*\* ID\*\*STRV10

G 10

SSSSSSSS	TTTTTTTTTT	RRRRRRRR	VV	VV	IIIIII	000000
SSSSSSSS	TTTTTTTTTT	RRRRRRRR	VV	VV	IIIIII	000000
SS	TT	RR	RR	VV	VV	OO
SS	TT	RR	RR	VV	VV	OO
SS	TT	RR	RR	VV	VV	OO
SS	TT	RR	RR	VV	VV	OO
SSSSSS	TT	RRRRRRRR	VV	VV	IIII	00
SSSSSS	TT	RRRRRRRR	VV	VV	IIII	00
SS	TT	RR	RR	VV	VV	OO
SS	TT	RR	RR	VV	VV	OO
SS	TT	RR	RR	VV	VV	OO
SS	TT	RR	RR	VV	VV	OO
SSSSSSSS	TT	RR	RR	VV	IIIIII	000000
SSSSSSSS	TT	RR	RR	VV	IIIIII	000000

```
1 0001 0
2 0002 0 MODULE STRVIO (LANGUAGE (BLISS32) .
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1 ****
7 0007 1 ****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 ****
29 0029 1 ++
30 0030 1 ++
31 0031 1 +
32 0032 1 FACILITY: MTAACP
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1 This module requeues block virtual IO on the current unit
36 0036 1
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 STARLET operating system, including privileged system services
41 0041 1 and internal exec routines.
42 0042 1 --
43 0043 1 --
44 0044 1
45 0045 1
46 0046 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 30-AUG-1977
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1 V03-002 MMD0002 Meg Dumont, 9-Feb-1983 15:45
51 0051 1 Clean up of START_VIO, so that no locations in paged
52 0052 1 memory are accessed while we are at FIPL. Also
53 0053 1 fix to always set the UCB address before calling
54 0054 1 EXESINSIOQ.
55 0055 1
56 0056 1 V03-001 MMD0001 Meg Dumont, 3-Jan-1983 16:18
57 0057 1 Added routine that is called when the MTAACP has read a trailer
```

```
58 0058 1 : label. Its purpose is to stop accidental reading or writing
59 0059 1 : of the trailer labels.
60 0060 1 :
61 0061 1 : V02-002 REFORMAT      Maria del C. Nasr      30-Jun-1980
62 0062 1 :
63 0063 1 : **
64 0064 1 :
65 0065 1 : LIBRARY 'SYSSLIBRARY:LIB.L32';
66 0066 1 :
67 0067 1 : REQUIRE 'SRC$:MTADEF.B32';
68 0451 1 :
69 0452 1 : LINKAGE
70 0453 1 :     INS_QUE      = JSB (REGISTER = 3, REGISTER = 5) : NOPRESERVE (1, 2, 4);
71 0454 1 :
72 0455 1 : EXTERNAL
73 0456 1 :     CURRENT_UCB : REF BBLOCK,          | address of current unit control block
74 0457 1 :     CURRENT_WCB : REF BBLOCK,          | address of current window control block
75 0458 1 :     QUEUE_HEAD : REF BBLOCK;          | address of ACP input queue head
76 0459 1 :
77 0460 1 : EXTERNAL ROUTINE
78 0461 1 :     EXE$INSIOQ : INS_QUE ADDRESSING_MODE (ABSOLUTE);
79 0462 1 :
80 0463 1 : LOCK_CODE:
81 0464 1 :
```

83 0465 1 GLOBAL ROUTINE START\_VIO : COMMON\_CALL NOVALUE =  
84 0466 1  
85 0467 1 ++  
86 0468 1  
87 0469 1 FUNCTIONAL DESCRIPTION:  
88 0470 1 This routine queues blocked virtual IO to the current unit  
89 0471 1  
90 0472 1 CALLING SEQUENCE:  
91 0473 1 START\_VIO(), called in kernel mode  
92 0474 1  
93 0475 1 INPUT PARAMETERS:  
94 0476 1 none  
95 0477 1  
96 0478 1 IMPLICIT INPUTS:  
97 0479 1 CURRENT\_UCB - address of current unit control block to which all  
98 0480 1 blocked virtual IO is to be queued  
99 0481 1 CURRENT\_VCB - address of current volume control block which contains  
100 0482 1 the blocked virtual IO list head  
101 0483 1 CURRENT\_WCB - address of current window control block which is currently  
102 0484 1 not mapping virtual IO  
103 0485 1 QUEUE\_HEAD - address of ACP input queue head to which mapping errors  
104 0486 1 for this volume are queued  
105 0487 1  
106 0488 1 OUTPUT PARAMETERS:  
107 0489 1 none  
108 0490 1  
109 0491 1 IMPLICIT OUTPUTS:  
110 0492 1 window restored mapping to current unit  
111 0493 1 virtual IO requeued to that unit  
112 0494 1  
113 0495 1 ROUTINE VALUE:  
114 0496 1 none  
115 0497 1  
116 0498 1 SIDE EFFECTS:  
117 0499 1 this routine runs at fork level in order to synchronize with the mapping  
118 0500 1 portion of QIO processing and the magnetic tape driver  
119 0501 1  
120 0502 1 --  
121 0503 1  
122 0504 2 BEGIN  
123 0505 2  
124 0506 2 EXTERNAL REGISTER  
125 0507 2 COMMON\_REG;  
126 0508 2  
127 0509 2 LOCAL  
128 0510 2 ENTRY : REF BBLOCK, ! address of ACP queue entry  
129 0511 2 FUNCTION, ! IO function code  
130 0512 2 PACKET : REF BBLOCK, ! address of IO request packet which is to be requeued  
131 0513 2 UCB : REF BBLOCK, ! address of current UCB  
132 0514 2 VCB : REF BBLOCK, ! address of current volume control block  
133 0515 2 WCB : REF BBLOCK, ! address of the window control block  
134 0516 2 QUEUE\_ENTRY : REF BBLOCK, ! address of ACP input queue  
135 0517 2 UCB\_SAVE : REF BBLOCK; ! address of saved UCB  
136 0518 2  
137 0519 2 ! All data structures used in these routine must be accessible from  
138 0520 2 local storage. Because we raise to fork IPL we can not afford to  
139 0521 2 cause a page fault.

```
140      0522 2
141      0523 2      UCB = .CURRENT_UCB;          | address of UCB to which io is to be queue
142      0524 2      VCB = .CURRENT_VCB;        | address of current VCB
143      0525 2      WCB = .CURRENT_WCB;        | address of current WCB
144      0526 2      QUEUE_ENTRY = .QUEUE_HEAD;    | address of ACP input queue
145      0527 2      UCB_SAVE = .CURRENT_UCB;     | Address of current UCB
146      0528 2      SET_IPL(.UCB[UCBS$B_FIPL]);   | raise to fork level
147      0529 2
148      0530 2      | fix up map pointer with UCB which is to receive virtual IO
149      0531 2
150      0532 2      WCB[WCBS$W_NMAP] = 1;
151      0533 2      (WCBS$W_P1_COUNT)<0, 32> = .UCB;
152      0534 2
153      0535 2      | requeue all blocked io to current unit
154      0536 2
155      0537 2
156      0538 2      WHILE 1
157      0539 2      DO
158      0540 3      BEGIN
159      0541 3
160      0542 3      IF REMQUE(.VCB[VCBS$L_BLOCKFL], PACKET)
161      0543 3      THEN
162      0544 3          EXITLOOP;
163      0545 3
164      0546 3      | may have been cleared when error was processed
165      0547 3
166      0548 3      PACKET[IRPSV_VIRTUAL] = 1;
167      0549 3
168      0550 3      ! this is here because INSIOQ does not preserve R5
169      0551 3
170      0552 3      UCB = .UCB_SAVE;
171      0553 3      EXE$INSIOQ7.PACKET, .UCB);
172      0554 2      END;
173      0555 2
174      0556 2      | Scan input queue for any mapping errors that belong to this volume
175      0557 2      | put them at the tail of blocked IO list. Also the MSCP tape drives
176      0558 2      | outstanding I/O's will be found in this queue under most circumstances.
177      0559 2
178      0560 2      ENTRY = .QUEUE_ENTRY[AQBS$L_ACPQFL];
179      0561 2
180      0562 2      WHILE .ENTRY NEQA .QUEUE_ENTRY
181      0563 2      DO
182      0564 3      BEGIN
183      0565 3      FUNCTION = .ENTRY[IRPSV_FCODE];
184      0566 3
185      0567 3      IF .FUNCTION EQL IOS_READPBLK
186      0568 3          OR
187      0569 3          .FUNCTION EQL IOS_WRITEPBLK
188      0570 3      THEN
189      0571 4      BEGIN
190      0572 4
191      0573 4      IF .BBLOCK[.ENTRY[IRPSL_UCB], UCBSL_VCB] EQLA .VCB
192      0574 4      THEN
193      0575 5      BEGIN
194      0576 5      ENTRY = .ENTRY[IRPSL_IOQBL];
195      0577 5      REMQUE(.ENTRY[IRPSL_IOQFL], PACKET);
196      0578 5
```

```

197      0579 5      ! may have been cleared when error was processed
198      0580 5
199      0581 5      PACKET[IRPSV_VIRTUAL] = 1;
200      0582 5      UCB = .UCB_SAVE;
201      0583 5      EXE$INSIOQT.PACKET, .UCB);
202      0584 4      END;
203      0585 4
204      0586 3      END;
205      0587 3
206      0588 3      ENTRY = .ENTRY[IRPSL_IOQFL];
207      0589 2      END;
208      0590 2
209      0591 2      SET_IPL(0);
210      0592 1      END;

```

```

.TITLE STRVIO
.IDENT \V04-000\

.EXTRN CURRENT_UCB, CURRENT_WCB
.EXTRN QUEUE_HEAD, EXE$INSIOQ
.PSECT $LOCKEDC1$,NOWRT,2

      07FC 00000
      55 0000G CF D0 00002
      5A 0000G 5B D0 00007
      50 0000G CF D0 0000A
      59 0000G CF D0 0000F
      58 0000G CF D0 00014
      51 0B A5 9A 00019
      12 51 DA 0001D
      16 A0 01 B0 00020
      30 A0 55 D0 00024
      53 00 BA OF 00028 1$: REMQUE
      0F 1D 0002C
      2A A3 10 88 0002E
      55 58 D0 00032
      00000000G 9F 16 00035
      EB 11 0003B
      56 69 D0 0003D 2$: MOVL
      59 56 D1 00040 3$: CMPL
      34 13 00043
      06 00 EF 00045
      0C 57 D1 0004B
      08 05 13 0004E
      1F 57 D1 00050
      50 1C A6 D0 00055 4$: BNEQ
      5A 34 A0 D1 00059
      15 12 0005D
      56 04 A6 D0 0005F
      53 00 B6 OF 00063
      2A A3 10 88 00067
      55 58 D0 0006B
      00000000G 9F 16 0006E
      56 66 D0 00074 5$: MOVL
      07FC 00000
      MOVL CURRENT_UCB, UCB
      MOVL CURRENT_VCB, VCB
      MOVL CURRENT_WCB, WCB
      MOVL QUEUE_HEAD, QUEUE_ENTRY
      MOVL CURRENT_UCB, UCB_SAVE
      MOVZBL 11(UCB), R1
      MTPR R1, #18
      MOVW #1, 22(WCB)
      MOVL UCB, 48(WCB)
      BVS 2$
      BISB2 #16, 42(PACKET)
      MOVL UCB_SAVE, UCB
      JSB @#EXE$INSIOQ
      BRB 1$
      (QUEUE_ENTRY), ENTRY
      ENTRY, QUEUE_ENTRY
      BEQL 6$
      EXTZV #0, #6, 32(ENTRY), FUNCTION
      CMPL FUNCTION, #12
      BEQL 4$
      CMPL FUNCTION, #11
      BNEQ 5$
      MOVL 28(ENTRY), R0
      CMPL 52(R0), VCB
      BNEQ 5$
      MOVL 4(ENTRY), ENTRY
      REMQUE @0(ENTRY), PACKET
      BISB2 #16, 42(PACKET)
      MOVL UCB_SAVE, UCB
      JSB @#EXE$INSIOQ
      MOVL (ENTRY), ENTRY
      0465
      0523
      0524
      0525
      0526
      0527
      0528
      0532
      0533
      0542
      0548
      0552
      0553
      0538
      0560
      0562
      0565
      0567
      0569
      0573
      0576
      0577
      0581
      0582
      0583
      0588
      0523
      0524
      0525
      0526
      0527
      0528
      0532
      0533
      0542
      0548
      0552
      0553
      0538
      0560
      0562
      0565
      0567
      0569
      0573
      0576
      0577
      0581
      0582
      0583
      0588

```

WT  
VO  
: 0562  
: 0591  
: 059212 C7 11 00077  
00 DA 00079 6\$: BRB 3\$  
04 0007C MTPR #0, #18  
RET

: Routine Size: 125 bytes, Routine Base: \$LOCKEDC1\$ + 0000

```

: 211      0593 1
: 212      0594 1 GLOBAL ROUTINE STOP_VIO : COMMON_CALL NOVALUE =
: 213      0595 1 ++
: 214      0596 1
: 215      0597 1 FUNCTIONAL DESCRIPTION:
: 216      0598 1 This routine blocks virtual IO from happening over the current WCB
: 217      0599 1
: 218      0600 1 CALLING SEQUENCE:
: 219      0601 1 STOP_VIO(), called in kernel mode
: 220      0602 1
: 221      0603 1 INPUT PARAMETERS:
: 222      0604 1 none
: 223      0605 1
: 224      0606 1 IMPLICIT INPUTS:
: 225      0607 1 CURRENT_WCB - address of current window control block which is currently
: 226      0608 1 not mapping virtual IO
: 227      0609 1
: 228      0610 1 OUTPUT PARAMETERS:
: 229      0611 1 none
: 230      0612 1
: 231      0613 1 IMPLICIT OUTPUTS:
: 232      0614 1 window mapping to current unit stopped
: 233      0615 1
: 234      0616 1 ROUTINE VALUE:
: 235      0617 1 none
: 236      0618 1
: 237      0619 1 SIDE EFFECTS:
: 238      0620 1 this routine runs at fork level in order to synchronize with the mapping
: 239      0621 1 portion of QIO processing and the magnetic tape driver
: 240      0622 1
: 241      0623 1 --
: 242      0624 1
: 243      0625 2 BEGIN
: 244      0626 2
: 245      0627 2 EXTERNAL REGISTER
: 246      0628 2 COMMON_REG;
: 247      0629 2
: 248      0630 2 CURRENT_WCB[WCB$W_NMAP] = 0;
: 249      0631 1 END;

```

50      0000G    CF    0000 00000	.ENTRY   STOP_VIO, Save nothing
16      A0      B4    00007	MOV <sub>L</sub> CURRENT_WCB, R0
04    0000A	CLRW      22(R0)
	RET

WT  
VO  
: 0594  
: 0630  
: 0631

: Routine Size: 11 bytes, Routine Base: \$LOCKEDC1\$ + 007D

STRVIO  
VO4-000

N 10  
16-Sep-1984 02:33:08  
14-Sep-1984 12:46:50 VAX-11 Bliss-32 V4.0-742  
[MTAACP.SRC]STRVIO.B32;1

Page 7  
(2)

: 250 0632 1  
: 251 0633 1 END  
: 252 0634 1  
: 253 0635 0 ELUDOM

#### PSECT SUMMARY

Name	Bytes	Attributes
\$LOCKEDC1\$	136	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

#### Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
\$_255\$DUA28:[SYSLIB]LIB.L32;1	18619	17	0	1000	00:01.8

#### COMMAND QUALIFIERS

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

Size: 136 code + 0 data bytes  
Run Time: 00:08.0  
Elapsed Time: 00:17.8  
Lines/CPU Min: 4756  
Lexemes/CPU-Min: 19707  
Memory Used: 99 pages  
Compilation Complete

0256 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

